

A Viterbi Decoding Program for DSN Telemetry System Analysis

B. Benjauthrit, B. D. L. Mulhall, and J. S. Wong,
DSN Systems Engineering Section

A computer program written in Fortran V for the Univac 1108 to simulate the Viterbi decoding algorithm is described together with its capabilities and preliminary simulation results.

I. Introduction

To increase the capability of deep space telecommunication systems, convolutional coding is being implemented with Viterbi decoding. The first flight project to use this capability will be the Mariner Jupiter/Saturn 1977 mission. Subsequent interplanetary exploration missions will also employ this capability. In verifying the design and evaluating the telecommunication systems performance (particularly the DSN Telemetry System) for these missions, computer programs are a cost-effective way of achieving results. This technique can also be used for monitoring performance of the Telemetry System.

This is a progress report on the development of a computer program which implements the Viterbi decoding algorithm for analysis purposes. The program was modularized so that minor modifications or additional features can be relatively simple to incorporate into the program. Actually, there are many computer programs (Refs. 1-4) written to implement the algorithm. However, they do not possess the required capabilities for analysis of DSN Telemetry System performance.

II. Program Structure

The program is written in Fortran V and implemented on JPL's Univac 1108 system.¹ The program is conversational and is accessible through remote terminals such as Execuports and Uniscopes. Figure 1 depicts a basic flow chart of the Viterbi decoding algorithm. An enlarged version of the algorithm is block diagrammed in Fig. 2. Since most of the blocks in the diagram are self-explanatory, only brief discussions follow. Detailed explanations and terminology may be found in Refs. 1-5.

The program accepts convolutionally coded input data. The branch metrics are computed in subroutine BMTAB. The summary of decoding results can be printed out for any desirable number of decoded bits through Block 7. This is to monitor proper functioning of the program and to safeguard against total loss of decoding statistics in the event of an 1108 failure. Out-of-sync information used in

¹ The program is based on a Fortran V program written at the Johnson Space Center (Ref. 4). Capability added to the program at JPL includes the ability to read test data tapes, perform node synchronization, and compute error burst statistics.

Block 8 is flagged in Block 28. Illegal input symbols are detected in Block 8 and are outputted by Block 32. In order to determine error statistics, uncorrupted symbols and input data bits are generated internally or entered externally in Block 14. Selection of decoded bits is accomplished in Block 15, together with update statistics and print outs. The size of error-free gaps is recorded in Block 18 and is passed to Block 17 where the statistics of burst errors are analyzed. The summary of burst errors is also printed out periodically in Block 17. The out-of-sync condition determined in Block 27 on the basis of the number of decoded bits and the number of normalizations is accumulated in Blocks 19 and 26, respectively. Total decoding statistics are printed at an end-of-file, or when an error is detected in the data file.

III. Program Capabilities

The program was written for the DSN rate 1/2 and rate 1/3, constraint length 7, 8-level quantization, convolutional codes. Since the program structure is rather general, minimum modifications need be made to accommodate other short constraint length convolutional codes.

Though only two methods of selecting a decoded bit, i.e., the best metric decision and the majority vote decision, are available in the program, other alternative methods can be easily added to the program. Note that the maximum memory path length in bits (the value of $m \cdot$

K in Block 13 of Fig. 2) is 72; it can be expanded when desired. Besides providing bit error rate (BER), symbol error rate (SER), for the above two methods of selecting decoded bits, the hard input data symbols, symbol error locations, hard input data bits, and decoded data bits are also available.

The statistics of burst errors are: The number of errors per burst, the burst length in bits, and the error-free gap length in bits. These results are averaged periodically and summarized at the end of each run. The distribution of burst errors is also recorded.

IV. Preliminary Simulation Results

The program was employed to process test data obtained from CTA 21, after having been converted from a 64-level quantization into an 8-level acceptable data format. The test conditions of the data and partial input data for the program are described in Fig. 3. The statistics of raw-symbol and decoded-bit errors are tabulated in Fig. 4. Note that the result of Fig. 3a was obtained separately.²

Though the amount of data processed was too small to be of any significance, the results obtained appear to agree within the same order of magnitude with those provided by the LV7015 decoder of LINKABIT (Ref. 6). Comparison is made in Fig. 5.

²From a program written by Dr. C.A. Greenhall

References

1. Heller, J. A. and Jacobs, I. M., "Viterbi Decoding for Satellite and Space Communication," *IEEE Trans. on Comm. Tech.*, Vol. COM-19, No. 5, October 1971, pp. 835-848.
2. Gilhausen, K. S., et al., "Coding Systems Study for High Data Rate Telemetry Links," N71-27786, LINKABIT Corp., San Diego, Calif., January 1971.
3. Layland, J., "Performance of Short Constraint Length Convolutional Codes and a Heuristic Code-Construction Algorithm," in *The Deep Space Network, Space Programs Summary 37-64*, Vol. II, pp. 40-44, Jet Propulsion Laboratory, Pasadena, Calif., August 1970.
4. Batson, B., et al., "Simulation Results for the Viterbi Decoding Algorithm," NASA TR R-396, November 1972.
5. Viterbi, A. J., "Convolutional Codes and Their Performance in Communication Systems," *IEEE Trans. on Comm. Tech.*, Vol. COM-19, No. 5, pp. 751-772, October 1971.
6. *The LV7015 Reference Manual*, LINKABIT Corp., San Diego, Calif., 1972.

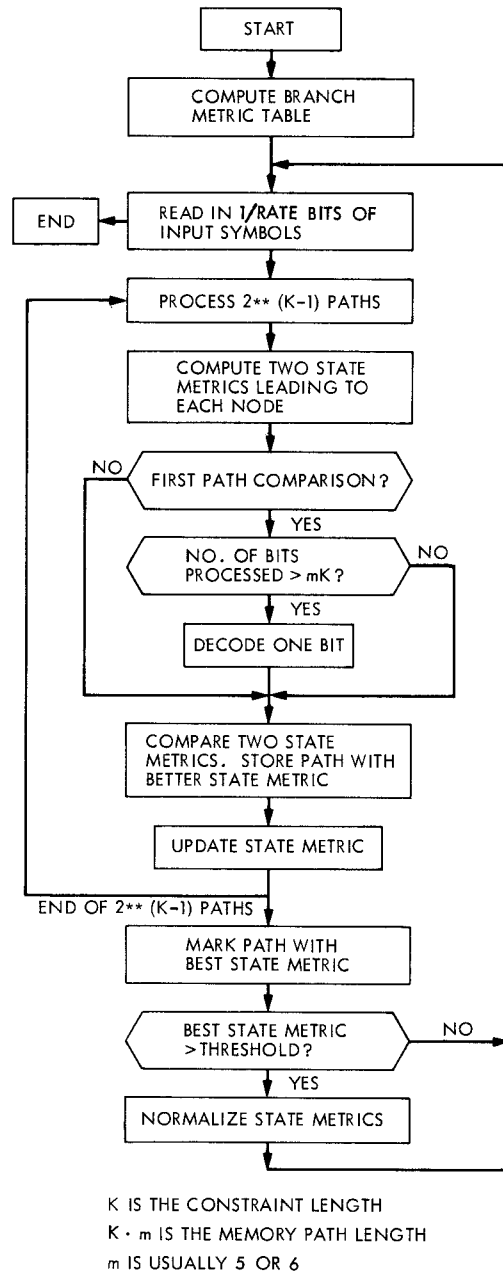


Fig. 1. Basic Viterbi decoding algorithm

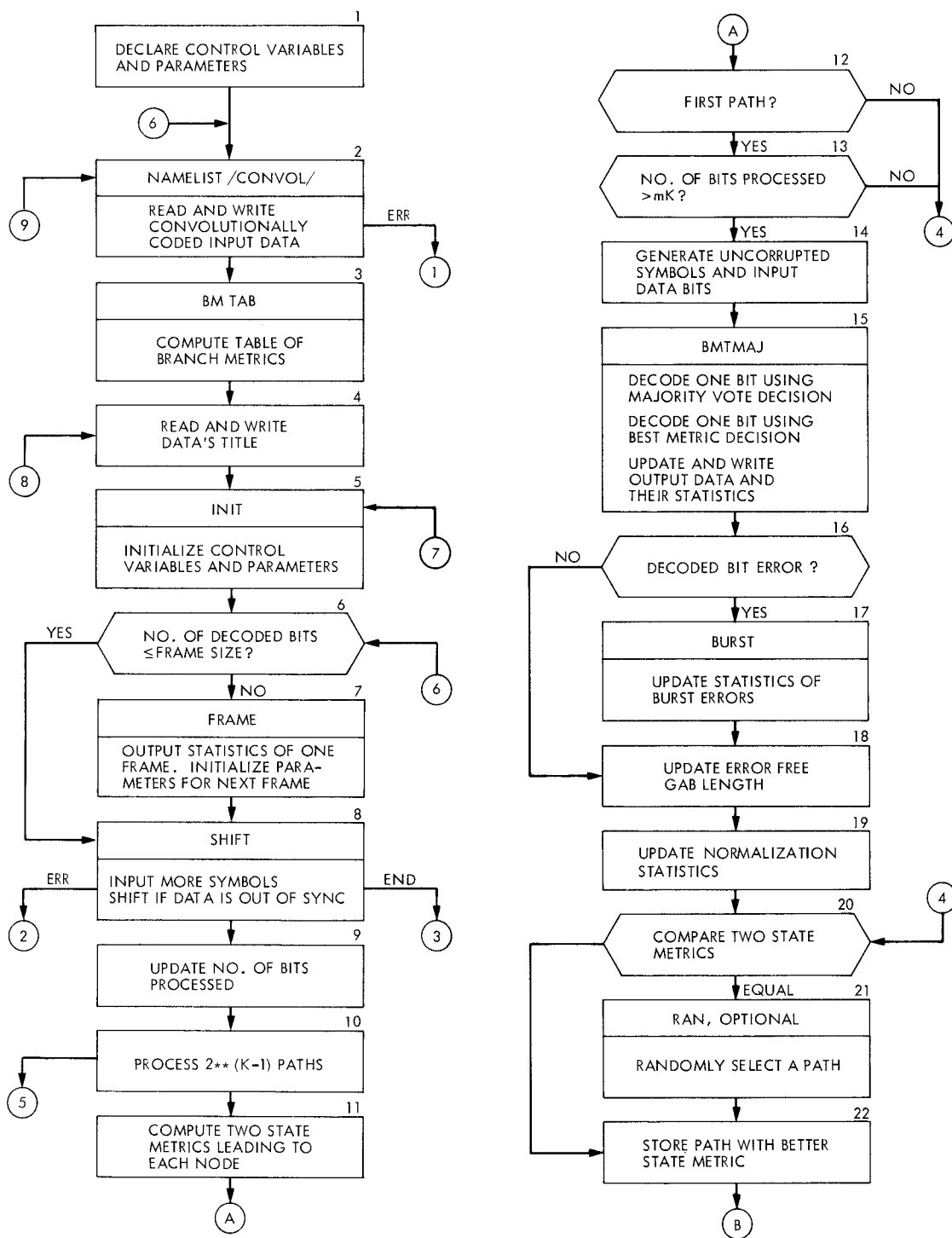


Fig. 2. Flow chart of 1108 program

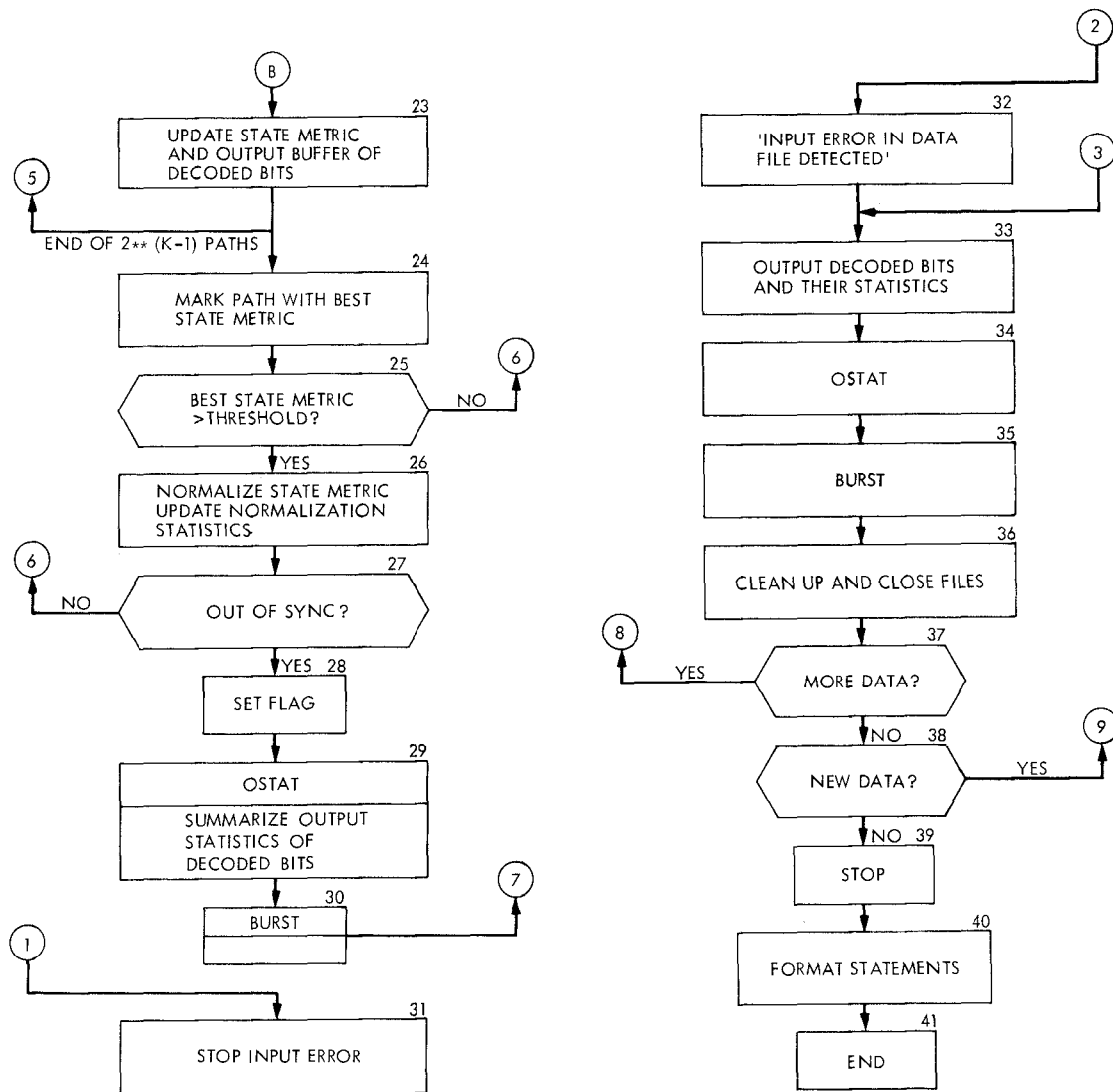


Fig. 2. (contd)

symbols/sec = 2048	E_b/N_0 , THEORETICAL, INCLUDING LOSSES = 3.715 dB
MODULATION INDEX = 55 deg	γ FACTOR = 6.29
CARRIER SUPPRESSION = -4.828 dB	PAD = 9.14
(a) TEST CONDITIONS	
RATE = 1/2	IMPULSE RESPONSE = 3133023 (QUATERNARY NOTATION)
CONSTRAINT LENGTH = 7	mK = 36
(b) PARTIAL INPUT TO 1108 PROGRAM	

Fig. 3. Test conditions and program input

	<u>NO. OF SYMBOLS</u>	<u>AVE. LENGTH IN BITS OF CONTINUOUS ERRORS</u>	<u>AVE. GAP LENGTH</u>
1.	9072	1.0756	14.8421
2.	6482	1.0772	14.5398
3.	3564	1.0602	12.2921

(a) STATISTIC OF RAW SYMBOLS

GAP LENGTH (IN bits)	6224	543	2331	45	6405	0	1	0	2641	4164	3240	4856	1626
	↑	↑	↑	↑	↑	↑	↑	↑	↑	↑	↑	↑	↑
ERROR LOCATION	1	2	3	4	5	6	7	8	9	10	11	12	

(b) BURST STATISTIC OF DECODED BITS

Fig. 4. Statistics of symbols and decoded bit errors

	<u>1108 PROGRAM</u>	<u>LV 7015*</u>
BER	3.74×10^{-4}	1.10×10^{-4}
SER	6.83×10^{-2}	—
AVE. NO. OF ERRORS/BURST**	4	3.8
AVE. NO. OF BIT BURST LENGTH	5	6.2
* FROM REF. 6 AT 4.0 dB		
** A BURST IS DEFINED HERE AS A SEQUENCE OF BIT ERRORS CONTAINING AT MOST K-2 GOOD BITS		

Fig. 5. Comparison of error statistics